Albert-Ludwigs-Universität
Institut für Informatik
Prof. Dr. Susanne Albers
Dr. Markus Schmidt
Sylva Scholz

Freiburg i. Br., den 24. Mai 2006

APPROXIMATIONS- UND ONLINE-ALGORITHMEN SS 06

Übung 3

Abgabe am 14. Juni 2006 (in der Vorlesung)

Aufgabe 1: Page Migration Problem (6 Punkte)
Beim *Page Migration* Problem sind eine Seite, ein ungerichteter Graph $G = (V, E)$ und ein Knoten, in dem die Seite anfangs gelagert ist, gegeben. Eine Anfrage besteht aus einem Knoten $x \in V$ und wird bearbeitet, indem eine Verbindung zum Knoten $y \in V$ hergestellt wird, in dem sich die Seite befindet. Die Kosten für diese Verbindung sind $d(x, y)$, die Länge des kürzesten Weges zwischen x und y. Im Anschluss an die Bedienung der Anfrage kann die Seite vom Knoten y in einen Knoten z verschoben werden. Hierbei entstehen Migrationskosten in Höhe von $D \cdot d(y, z)$.
Wir betrachten den Algorithmus RPM, der bei einer Anfrage in x zunächst die Anfrage bedient und dann die Seite mit einer Wahrscheinlichkeit $\frac{1}{2D}$ von y nach x verschiebt. Zeigen Sie mittels Potentialmethode, dass RPM 3-kompetitiv gegen den adaptiven Online-Gegner ist. Verwenden Sie hierbei die Potentialfunktion $\Phi = 3D \cdot d(s, t)$, mit s bzw. t als dem Knoten, bei dem der Gegner bzw. RPM die Seite gelagert hat.

Aufgabe 2: Datenkompression (3 + 3 Punkte)

a) Verschlüsseln Sie mit Hilfe der Burrows-Wheeler-Transformation das Wort *mississippi*.

b) Entschlüsseln Sie mit Hilfe der Burrows-Wheeler-Transformation das Wort *radgggeenee* mit Index 7.

Aufgabe 3: Online Routing (3 + 3 Punkte)
Gegeben ist ein ungerichtetes Netzwerk $G = (V, E, u)$, mit V als der Knotenmenge, E als der Kantenmenge und $u : E \to \mathbb{R}_{\geq 0}$ als Kapazitätsfunktion. Eine Anfrage $r_j = (s_j, t_j, b_j)$ ist die Aufforderung, eine Verbindung zwischen den Knoten s_j und t_j mit Bandbreite b_j herzustellen. Wird diese Verbindung auf dem Pfad P in G geroutet, erhöht sich die Last auf jeder Kante von P um b_j. Eine Anfrage kann angenommen werden, wenn jede Kante von P noch b_j freie Kapazität besitzt. Ziel ist es, die Summe der Bandbreiten der akzeptierten Anfragen zu maximieren.
Wir betrachten den Spezialfall, in dem $G = (V, E)$ ein Pfad aus N Knoten ist mit $u(e) = 1$ für alle Kanten $e \in E$. Für jede Anfrage r_j gilt $b_j = 1$. Somit soll die Anzahl kantendisjunkter Wege in G mit den entsprechenden Endpunkten aus den akzeptierten Anfragen maximiert werden.

a) Zeigen Sie, dass jeder deterministische Online-Algorithmus auf G nicht besser als $(N-1)$-kompetitiv ist.

b) Sei $N = 2^p$. Es bezeichne e_1 diejenige Kante, durch deren Entfernung G in zwei Pfade G' und G'' zu je 2^{p-1} Knoten zerfällt. Wir setzen $E_1 = \{e_1\}$. Die Pfade G' und G'' werden ebenfalls jeweils in der Mitte geteilt. Seien e_{21}, e_{22} die entsprechenden Kanten und $E_2 = \{e_{21}, e_{22}\}$. Rekursive Fortsetzung liefert die Partition von E in Mengen $E_1, \ldots, E_p$.
Der Algorithmus RAND wählt zufällig und gleichverteilt ein $l \in \{1, 2, \ldots, p\}$ und akzeptiert nur Anfragen $r_j = (s_j, t_j, 1)$ mit $l = \min\{i : E_i \cap E' \neq \emptyset\}$, wobei E' die Kanten auf dem Weg zwischen s_j und t_j sind. Zeigen Sie, dass RAND $\log_2 N$-kompetitiv ist.
(Hinweis: Seien R_k und O_k die Anzahl der von RAND und einem optimalen Offline-Algorithmus OPT akzeptierten Anfragen mit $k = \min\{i : E_i \cap E' \neq \emptyset\}$. Zeigen Sie zunächst, dass $R_l \geq O_l$ gilt.)

Bemerkung: Bei der kompetitiven Analyse werden keine additiven Konstanten zugelassen, da sonst mit $a := |E|$ jeder Algorithmus 1-kompetitiv wäre.

Aufgabe 4: Robotik (6 Punkte)

Wir betrachten einen Roboter, der auf einem rechteckigen Gitter in der Ebene ausgehend von einem Startpunkt s einen Punkt finden soll. Der Abstand der Punkte auf einer Linie des Gitters beträgt 1 LE. Der Roboter kann sich pro Zug um 1 LE nach Norden, Süden, Westen oder Osten bewegen. Punkte, die $k \leq n$ Schritte von s entfernt liegen, beschreiben den n*-ten Diamanten* um s. Sei ALG ein Algorithmus, der die Suche des Roboters bestimmt, und $A(n)$ die Anzahl der Schritte, die ALG benötigt um alle Punkte im n-ten Diamanten zu besuchen. Zeigen Sie, dass für jeden Algorithmus ALG gilt:

$$A(n) \geq 2\,n^2 + 4\,n + 1 \text{ oder } A(n+1) \geq 2\,(n+1)^2 + 4\,(n+1) + 1.$$